Hoare logic: proof-tree style

1. Example proof of:

$$
\begin{array}{l}
\{\ \} \\
y := 0\,; \\
z := 1\,; \\
\texttt{while}\ y \neq x\ \texttt{do} \\
\qquad y := y + 1\,; \\
\qquad z := z * y \\
\{\, z = x!\, \}
\end{array}
$$

2. The proof rules.

Abbreviations:

$$W: \quad \texttt{while } y \neq x \texttt{ do } y := y+1\,;\, z := z * y$$

$$P_{\text{fact}}: \quad y := 0\,;\, z := 1\,;\, W$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\{\, z*(y+1)=(y+1)!,\ y+1 \geq 0 \,\}\ y:=y+1\ \{\, z*y=y!,\ y \geq 0 \,\}}}{\{\, z=y!,\ y \geq 0,\ y \neq x \,\}\ y:=y+1\ \{\, z*y=y!,\ y \geq 0 \,\}} \qquad \dfrac{}{\{\, z*y=y!,\ y \geq 0 \,\}\ z:=z*y\ \{\, z=y!,\ y \geq 0 \,\}}}{\{\, z=y!,\ y \geq 0,\ y \neq x \,\}\ y := y+1\,;\ z := z*y\ \{\, z=y!,\ y \geq 0 \,\}}}{\{\, z=y!,\ y \geq 0 \,\}\ W\ \{\, z=y!,\ y \geq 0,\ y=x \,\}}}{\{\, z=y!,\ y \geq 0 \,\}\ W\ \{\, z=x! \,\}}$$

$$\dfrac{\dfrac{\dfrac{}{\{\ \}\ y:=0\ \{\, y!=1,\ y \geq 0 \,\}} \qquad \dfrac{}{\{\, y!=1,\ y \geq 0 \,\}\ z:=1\ \{\, z=y!,\ y \geq 0 \,\}}}{\{\ \}\ y := 0\,;\ z := 1\ \{\, z=y!,\ y \geq 0 \,\}} \qquad \begin{array}{c}\text{(proof above)}\\ \vdots \\ \{\, z=y!,\ y \geq 0 \,\}\ W\ \{\, z=x! \,\}\end{array}}{\{\ \}\ P_{\text{fact}}\ \{\, z=x! \,\}}$$

Proof rules

$$\frac{\{\,\eta,\ B\,\}\ C\ \{\,\eta\,\}}{\{\,\eta\,\}\ \texttt{while}\ B\ \texttt{do}\ C\ \{\,\eta,\ \neg B\,\}}\ \text{(partial while)}$$

$$\frac{\{\,\phi\,\}\ C_1\ \{\,\eta\,\} \quad \{\,\eta\,\}\ C_2\ \{\,\psi\,\}}{\{\,\phi\,\}\ C_1\ ;\ C_2\ \{\,\psi\,\}}\ \text{(composition)}$$

$$\frac{}{\{\,\phi[E/x]\,\}\ x := E\ \{\,\phi\,\}}\ \text{(assignment)}$$

$$\frac{}{\{\,\phi\,\}\ \texttt{skip}\ \{\,\phi\,\}}\ \text{(skip)}$$

$$\frac{\{\,\eta,\, B\,\}\; C_1\; \{\,\psi\,\} \qquad \{\,\eta,\, \neg B\,\}\; C_2\; \{\,\psi\,\}}{\{\,\eta\,\}\; \texttt{if } B \texttt{ then } C_1 \texttt{ else }`C_2\; \{\,\psi\,\}} \text{ (conditional)}$$

$$\frac{\{\,\phi'\,\}\; C\; \{\,\psi'\,\}}{\{\,\phi\,\}\; C\; \{\,\psi\,\}} \text{ (consequence)}^*$$

* The side-condition for the consequence rule is that the implications $\phi \to \phi'$ and $\psi' \to \psi$ both express true properties of the integers; i.e.,

$$\mathbb{Z} \models \phi \to \phi' \quad \text{and} \quad \mathbb{Z} \models \psi' \to \psi$$

Tableaux rules

$\{\,\psi[E/x]\,\}$
$x := E$
$\{\,\psi\,\}$ assignment

$\{\,\psi\,\}$
`skip`
$\{\,\psi\,\}$ skip

$\{\,\eta\,\}$
`while` B `do`
  $\{\,\eta,\ B\,\}$ do precondition
  C
  $\{\,\eta\,\}$
$\{\,\eta,\ \neg B\,\}$ partial while

$\{\,\phi\,\}$
$\{\,\psi\,\}$ implied

(if $\mathbb{Z} \models \phi \rightarrow \psi$)

$\{\, \phi \,\}$
`if` B `then`
    $\{\, \phi, B \,\}$ then precondition
    C_1
    $\{\, \psi \,\}$
`else`
    $\{\, \phi, \neg B \,\}$ else precondition
    C_2
    $\{\, \psi \,\}$
$\{\, \psi \,\}$ if statement

Example tableaux-style proof

```
while x ≠ y do
    if x < y then
        y := y − x
    else
        x := x − y
```

Example tableaux-style proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$ precondition

$\texttt{while}\ x \neq y\ \texttt{do}$

$\quad \texttt{if}\ x < y\ \texttt{then}$

$\qquad y := y - x$

$\quad \texttt{else}$

$\qquad x := x - y$

$\{\, x = \mathrm{gcd}(x_0, y_0) \,\}$

Example tableaux-style proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$ precondition
$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0) \,\}$
`while` $x \neq y$ `do`
    $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x \neq y \,\}$ do precondition
    `if` $x < y$ `then`

        $y := y - x$

    `else`

        $x := x - y$

    $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0) \,\}$
$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x = y \,\}$ partial while
$\{\, x = \gcd(x_0, y_0) \,\}$ implied

Example tableaux-style proof

$$
\begin{array}{ll}
\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\} & \text{precondition} \\
\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\} & \\
\texttt{while } x \neq y \texttt{ do} & \\
\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x \neq y \,\} & \text{do precondition} \\
\quad \texttt{if } x < y \texttt{ then} & \\
\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x < y \,\} & \text{then precondition} \\
& \\
\qquad y := y - x & \\
\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\} & \\
\quad \texttt{else} & \\
\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x > y \,\} & \text{else precondition} \\
& \\
\qquad x := x - y & \\
\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\} & \\
\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\} & \text{if statement} \\
\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x = y \,\} & \text{partial while} \\
\{\, x = \mathrm{gcd}(x_0, y_0) \,\} & \text{implied}
\end{array}
$$

Example tableaux-style proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$	precondition
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	
`while` $x \neq y$ `do`	
$\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x \neq y \,\}$	do precondition
$\quad$ `if` $x < y$ `then`	
$\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x < y \,\}$	then precondition
$\qquad y := y - x$	
$\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	
$\quad$ `else`	
$\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x > y \,\}$	else precondition
$\qquad \{\, x > y > 0, \mathrm{gcd}(x - y, y) = \mathrm{gcd}(x_0, y_0) \,\}$	
$\qquad x := x - y$	
$\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	assignment
$\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	if statement
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x = y \,\}$	partial while
$\{\, x = \mathrm{gcd}(x_0, y_0) \,\}$	implied

Example tableaux-style proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$	precondition
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	
`while` $x \neq y$ `do`	
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x \neq y \,\}$	do precondition
`if` $x < y$ `then`	
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x < y \,\}$	then precondition
$y := y - x$	
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	
`else`	
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x > y \,\}$	else precondition
$\{\, x > y > 0, \mathrm{gcd}(x - y, y) = \mathrm{gcd}(x_0, y_0) \,\}$	implied
$x := x - y$	
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	assignment
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	if statement
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x = y \,\}$	partial while
$\{\, x = \mathrm{gcd}(x_0, y_0) \,\}$	implied

Example tableaux-style proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$	precondition
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	
`while` $x \neq y$ `do`	
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x \neq y \,\}$	do precondition
`if` $x < y$ `then`	
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x < y \,\}$	then precondition
$\{\, y > x > 0, \mathrm{gcd}(x, y - x) = \mathrm{gcd}(x_0, y_0) \,\}$	
$y := y - x$	
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	assignment
`else`	
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x > y \,\}$	else precondition
$\{\, x > y > 0, \mathrm{gcd}(x - y, y) = \mathrm{gcd}(x_0, y_0) \,\}$	implied
$x := x - y$	
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	assignment
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	if statement
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x = y \,\}$	partial while
$\{\, x = \mathrm{gcd}(x_0, y_0) \,\}$	implied

Example tableaux-style proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$	precondition
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	
`while` $x \neq y$ `do`	
$\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x \neq y \,\}$	do precondition
$\quad$ `if` $x < y$ `then`	
$\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x < y \,\}$	then precondition
$\qquad \{\, y > x > 0, \mathrm{gcd}(x, y - x) = \mathrm{gcd}(x_0, y_0) \,\}$	implied
$\qquad y := y - x$	
$\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	assignment
$\quad$ `else`	
$\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x > y \,\}$	else precondition
$\qquad \{\, x > y > 0, \mathrm{gcd}(x - y, y) = \mathrm{gcd}(x_0, y_0) \,\}$	implied
$\qquad x := x - y$	
$\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	assignment
$\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	if statement
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x = y \,\}$	partial while
$\{\, x = \mathrm{gcd}(x_0, y_0) \,\}$	implied

Example tableaux-style proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$	precondition
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	implied
`while` $x \neq y$ `do`	
$\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x \neq y \,\}$	do precondition
$\quad$ `if` $x < y$ `then`	
$\quad\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x < y \,\}$	then precondition
$\quad\quad \{\, y > x > 0, \mathrm{gcd}(x, y - x) = \mathrm{gcd}(x_0, y_0) \,\}$	implied
$\quad\quad y := y - x$	
$\quad\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	assignment
$\quad$ `else`	
$\quad\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x > y \,\}$	else precondition
$\quad\quad \{\, x > y > 0, \mathrm{gcd}(x - y, y) = \mathrm{gcd}(x_0, y_0) \,\}$	implied
$\quad\quad x := x - y$	
$\quad\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	assignment
$\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0) \,\}$	if statement
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x = y \,\}$	partial while
$\{\, x = \mathrm{gcd}(x_0, y_0) \,\}$	implied

The four implications needed in the previous proof are:

$$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\} \\ \Rightarrow \{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0) \,\}$$

$$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x < y \,\} \\ \Rightarrow \{\, y > x > 0, \gcd(x, y - x) = \gcd(x_0, y_0) \,\}$$

$$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x > y \,\} \\ \Rightarrow \{\, x > y > 0, \gcd(x - y, y) = \gcd(x_0, y_0) \,\}$$

$$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x = y \,\} \\ \Rightarrow \{\, x = \gcd(x_0, y_0) \,\}$$

The first and last are trivial.

The second and third are simple propositions in number theory.

Soundness for partial correctness

$\models_{\text{par}} \{\phi\}\, C\, \{\psi\} \iff$ for every state s satisfying ϕ,
if the execution of C from s terminates
then it terminates in a state satisfying ψ

$\vdash_{\text{par}} \{\phi\}\, C\, \{\psi\} \iff$ there exists a proof of $\{\phi\}\, C\, \{\psi\}$ using
the proof rules for partial correctness

Theorem (Soundness). If $\vdash_{\text{par}} \{\phi\} C \{\psi\}$ then $\models_{\text{par}} \{\phi\} C \{\psi\}$.
(Every provable formula is true)

Soundness is proved by showing that each inference rule preserves partial correctness.

The lemma on the next slide establishes this preservation property for the partial-while rule, which is the most interesting case.

Lemma. If $\models_{\text{par}} \{\eta, B\} C \{\eta\}$ then $\models_{\text{par}} \{\eta\}$ `while` B `do` $C \{\eta, \neg B\}$.

Proof. We prove by induction on n that, for every s satisfying η, if the execution of `while` B `do` C from s terminates after n iterations of the C loop then it terminates in a state satisfying $\eta \wedge \neg B$.

In the case that B is false in state s, the execution of the while loop aborts immediately, terminating in state s itself. By assumption, s indeed satisfies $\eta \wedge \neg B$. This establishes the case $n = 0$.

... continued on next slide

In the case that B is true in state s, the execution of the while loop proceeds as follows. First the command C is executed. If the execution of C terminates in some state s', then the main while loop is executed again from state s'.

Suppose the execution of `while` B `do` C from s terminates after n iterations of the C loop. Then $n > 0$, the execution of C from s terminates in some state s', and the execution of `while` B `do` C from s' terminates after $n - 1$ further iterations of the C loop.

By assumption, s satisfies $\eta \wedge B$. Because $\models_{\mathsf{par}} \{\, \eta, B \,\} C \{\, \eta \,\}$, the state s' satisfies η. By induction hypothesis, the state s'' resulting from the execution of `while` B `do` C from s' satisfies $\eta \wedge \neg B$.

But s'' is the state resulting from the execution of `while` B `do` C from s. This state indeed satisfies $\eta \wedge \neg B$ as required. $\square$

Hoare logic: total correctness

Proof rule:

$$\frac{\{\,\eta,\ B,\ 0 \leq E = z_0\,\}\ C\ \{\,\eta,\ 0 \leq E < z_0\,\}}{\{\,\eta,\ 0 \leq E\,\}\ \texttt{while}\ B\ \texttt{do}\ C\ \{\,\eta,\ \neg B\,\}}\ \text{(total while)}$$

z_0 is required to be a fresh variable.

- Property η is called the invariant for the while loop.
- Expression E is called the variant for the while loop.

Hoare logic: total correctness

Tableaux rule:

$$
\begin{array}{ll}
\{\,\eta,\ 0 \le E\,\} & \\
\texttt{while } B \texttt{ do} & \\
\qquad \{\,\eta,\ B,\ 0 \le E = z_0\,\} & \text{do precondition} \\
\qquad C & \\
\qquad \{\,\eta,\ 0 \le E < z_0\,\} & \\
\{\,\eta,\ \neg B\,\} & \text{total while}
\end{array}
$$

Soundness for total correctness

$\models_{\text{tot}} \{\,\phi\,\}\, C\, \{\,\psi\,\} \;\Leftrightarrow$ the execution of C from any state satisfying ϕ terminates in a state satisfying ψ

$\vdash_{\text{tot}} \{\,\phi\,\}\, C\, \{\,\psi\,\} \;\Leftrightarrow$ there exists a proof of $\{\,\phi\,\}\, C\, \{\,\psi\,\}$ using the proof rules for total correctness

Theorem (Soundness). If $\vdash_{\text{tot}} \{\,\phi\,\}C\{\,\psi\,\}$ then $\models_{\text{tot}} \{\,\phi\,\}C\{\,\psi\,\}$.

Soundness is proved by showing that each inference rule preserves total correctness.

The lemma on the next slide establishes this preservation property for the total-while rule, which is the most interesting case.

Lemma. If $\models_{\mathrm{tot}} \{\, \eta,\ B,\ 0 \le E = z_0 \,\}\, C \,\{\, \eta,\ 0 \le E < z_0 \,\}$ then $\models_{\mathrm{tot}} \{\, \eta,\ 0 \le E \,\}$ `while` B `do` $C \,\{\, \eta,\ \neg B \,\}$.

Proof. We prove by induction on n that, if `while` B `do` C is executed from any state s satisfying $\eta \wedge 0 \le E$, with $E^{s} = n$, then execution terminates in a state satisfying $\eta \wedge \neg B$. As induction hypothesis, we can assume this is true for every $n' < n$.

In the case that B is false in state s, the execution of the while loop aborts immediately, terminating in state s itself. By assumption, s indeed satisfies $\eta \wedge \neg B$.

... continued on next slide

In the case that B is true in state s, the execution of the while loop proceeds as follows. First the command C is executed. If the execution of C terminates in some state s', then the main while loop is executed again from state s'.

As z_0 is fresh, the state $s[z_0 \mapsto n]$ satisfies $\eta \wedge B \wedge 0 \leq E = z_0$. Because $\models_{\mathrm{tot}} \{\, \eta, B, 0 \leq E = z_0 \,\}\ C\ \{\, \eta, 0 \leq E < z_0 \,\}$, the execution of C from state $s[z_0 \mapsto n]$ terminates in some state s'' satisfying $\eta \wedge 0 \leq E < z_0$. So $E^{\,s''} = n'$ for some $n' < n$.

Since C does not contain z_0, the execution of C from s is the same as its execution from $s[z_0 \mapsto n]$, and thus terminates in a state s' such that $s'[z_0 \mapsto n] = s''$. Since E does not contain z_0, we have $E^{\,s'} = E^{\,s''} = n' < n$.

Having executed C to reach state s', the command `while` B `do` C is executed from state s'. Since s' satisfies $\eta \wedge 0 \leq E$, where $E^{\,s'} = n' < n$, the induction hypothesis yields that execution indeed terminates in a state satisfying $\eta \wedge \neg B$, as required. $\square$

Example total correctness proof

while $x \neq y$ **do**

  if $x < y$ **then**

    $y := y - x$

  else

    $x := x - y$

Example total correctness proof

$\{ x, y > 0,\ x = x_0,\ y = y_0 \}$ precondition

```
while x ≠ y do
    if x < y then

        y := y − x

    else

        x := x − y
```

$\{ x = \mathrm{gcd}(x_0, y_0) \}$

Example total correctness proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$ precondition
$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \leq x + y \,\}$
`while` $x \neq y$ `do`
  $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x \neq y, 0 \leq x + y = z_0 \,\}$ do precondition
  `if` $x < y$ `then`

    $y := y - x$

  `else`

    $x := x - y$

  $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \leq x + y < z_0 \,\}$
$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x = y \,\}$ total while
$\{\, x = \gcd(x_0, y_0) \,\}$ implied

Example total correctness proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$ precondition
$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \leq x + y \,\}$
`while` $x \neq y$ `do`
$\quad \{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x \neq y, 0 \leq x + y = z_0 \,\}$ do precondition
$\quad$ `if` $x < y$ `then`
$\quad\quad \{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x < y, 0 \leq x + y = z_0 \,\}$ then precondition

$\quad\quad y := y - x$
$\quad\quad \{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \leq x + y < z_0 \,\}$
$\quad$ `else`
$\quad\quad \{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x > y, 0 \leq x + y = z_0 \,\}$ else precondition

$\quad\quad x := x - y$
$\quad\quad \{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \leq x + y < z_0 \,\}$
$\quad \{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \leq x + y < z_0 \,\}$ if statement
$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x = y \,\}$ total while
$\{\, x = \gcd(x_0, y_0) \,\}$ implied

Example total correctness proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$ precondition
$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \le x + y \,\}$
`while` $x \neq y$ `do`
  $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x \neq y, 0 \le x + y = z_0 \,\}$ do precondition
  `if` $x < y$ `then`
    $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x < y, 0 \le x + y = z_0 \,\}$ then precondition

    $y := y - x$
    $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \le x + y < z_0 \,\}$
  `else`
    $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x > y, 0 \le x + y = z_0 \,\}$ else precondition
    $\{\, x > y > 0, \gcd(x - y, y) = \gcd(x_0, y_0), 0 \le x < z_0 \,\}$
    $x := x - y$
    $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \le x + y < z_0 \,\}$ assignment
  $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \le x + y < z_0 \,\}$ if statement
$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x = y \,\}$ total while
$\{\, x = \gcd(x_0, y_0) \,\}$ implied

Example total correctness proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$	precondition
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x + y \,\}$	
`while` $x \neq y$ `do`	
$\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x \neq y, 0 \leq x + y = z_0 \,\}$	do precondition
$\quad$ `if` $x < y$ `then`	
$\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x < y, 0 \leq x + y = z_0 \,\}$	then precondition
$\qquad y := y - x$	
$\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x + y < z_0 \,\}$	
$\quad$ `else`	
$\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x > y, 0 \leq x + y = z_0 \,\}$	else precondition
$\qquad \{\, x > y > 0, \mathrm{gcd}(x - y, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x < z_0 \,\}$	implied
$\qquad x := x - y$	
$\qquad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x + y < z_0 \,\}$	assignment
$\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x + y < z_0 \,\}$	if statement
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x = y \,\}$	total while
$\{\, x = \mathrm{gcd}(x_0, y_0) \,\}$	implied

Example total correctness proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$ precondition
$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \leq x + y \,\}$
`while` $x \neq y$ `do`
  $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x \neq y, 0 \leq x + y = z_0 \,\}$ do precondition
  `if` $x < y$ `then`
    $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x < y, 0 \leq x + y = z_0 \,\}$ then precondition
    $\{\, y > x > 0, \gcd(x, y - x) = \gcd(x_0, y_0), 0 \leq y < z_0 \,\}$
    $y := y - x$
    $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \leq x + y < z_0 \,\}$ assignment
  `else`
    $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x > y, 0 \leq x + y = z_0 \,\}$ else precondition
    $\{\, x > y > 0, \gcd(x - y, y) = \gcd(x_0, y_0), 0 \leq x < z_0 \,\}$ implied
    $x := x - y$
    $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \leq x + y < z_0 \,\}$ assignment
  $\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), 0 \leq x + y < z_0 \,\}$ if statement
$\{\, x, y > 0, \gcd(x, y) = \gcd(x_0, y_0), x = y \,\}$ total while
$\{\, x = \gcd(x_0, y_0) \,\}$ implied

Example total correctness proof

$\{\, x, y > 0,\ x = x_0,\ y = y_0 \,\}$	precondition
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x + y \,\}$	
`while` $x \neq y$ `do`	
$\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x \neq y, 0 \leq x + y = z_0 \,\}$	do precondition
$\quad$ `if` $x < y$ `then`	
$\quad\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x < y, 0 \leq x + y = z_0 \,\}$	then precondition
$\quad\quad \{\, y > x > 0, \mathrm{gcd}(x, y - x) = \mathrm{gcd}(x_0, y_0), 0 \leq y < z_0 \,\}$	implied
$\quad\quad y := y - x$	
$\quad\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x + y < z_0 \,\}$	assignment
$\quad$ `else`	
$\quad\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x > y, 0 \leq x + y = z_0 \,\}$	else precondition
$\quad\quad \{\, x > y > 0, \mathrm{gcd}(x - y, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x < z_0 \,\}$	implied
$\quad\quad x := x - y$	
$\quad\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x + y < z_0 \,\}$	assignment
$\quad \{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x + y < z_0 \,\}$	if statement
$\{\, x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x = y \,\}$	total while
$\{\, x = \mathrm{gcd}(x_0, y_0) \,\}$	implied

Example total correctness proof

$\{ x, y > 0,\ x = x_0,\ y = y_0 \}$	precondition
$\{ x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x + y \}$	implied
`while` $x \neq y$ `do`	
$\{ x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x \neq y, 0 \leq x + y = z_0 \}$	do precondition
`if` $x < y$ `then`	
$\{ x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x < y, 0 \leq x + y = z_0 \}$	then precondition
$\{ y > x > 0, \mathrm{gcd}(x, y - x) = \mathrm{gcd}(x_0, y_0), 0 \leq y < z_0 \}$	implied
$y := y - x$	
$\{ x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x + y < z_0 \}$	assignment
`else`	
$\{ x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x > y, 0 \leq x + y = z_0 \}$	else precondition
$\{ x > y > 0, \mathrm{gcd}(x - y, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x < z_0 \}$	implied
$x := x - y$	
$\{ x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x + y < z_0 \}$	assignment
$\{ x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), 0 \leq x + y < z_0 \}$	if statement
$\{ x, y > 0, \mathrm{gcd}(x, y) = \mathrm{gcd}(x_0, y_0), x = y \}$	total while
$\{ x = \mathrm{gcd}(x_0, y_0) \}$	implied

Completeness for partial correctness

Theorem (Completeness). If $\models_{\mathsf{par}} \{\,\phi\,\}C\{\,\psi\,\}$ then $\vdash_{\mathsf{par}} \{\,\phi\,\}C\{\,\psi\,\}$.

This theorem is due to Stephen A. Cook.

It is often called a relative completeness result because it is relative to an assumed external proof system for establishing the side-conditions of the consequence (implication) rule. Since the side conditions have the form $\mathbb{Z} \models \phi \to \psi$, they are ordinary mathematical statements.

By Gödel's celebrated incompleteness theorem for arithmetic, in reality any such external proof system is necessarily incomplete.

We outline the proof of the completeness theorem.

For every command C and assertion ψ, define $\mathsf{wp}(C, \psi)$ by:

$$s \text{ satisfies } \mathsf{wp}(C, \psi) \iff \begin{array}{l}\text{if the execution of } C \text{ from } s \text{ terminates} \\ \text{then the resulting state satisfies } \psi\end{array}$$

Lemma (expressive completeness). The property $\mathsf{wp}(C, \psi)$ can be expressed by a formula in our assertion logic.

Lemma (weakest precondition).

1. $\models_{\mathsf{par}} \{\, \mathsf{wp}(C, \psi) \,\}\, C \,\{\, \psi \,\}$.
2. If $\models_{\mathsf{par}} \{\, \phi \,\}\, C \,\{\, \psi \,\}$ then $\mathbb{Z} \models \phi \to \mathsf{wp}(C, \psi)$.

Lemma (sequencing).

1. $\mathbb{Z} \models \mathsf{wp}(C_1; C_2, \psi) \leftrightarrow \mathsf{wp}(C_1, \mathsf{wp}(C_2, \psi))$.
2. If $\models_{\mathsf{par}} \{\, \phi \,\}\, C_1 \,;\, C_2 \,\{\, \psi \,\}$ then $\models_{\mathsf{par}} \{\, \phi \,\}\, C_1 \,\{\, \mathsf{wp}(C_2, \psi) \,\}$.

Proof of completeness. We prove, by induction on the structure of commands C, that, for all assertions ϕ, ψ, it holds that $\models_{\mathsf{par}} \{\,\phi\,\} C \{\,\psi\,\}$ implies $\vdash_{\mathsf{par}} \{\,\phi\,\} C \{\,\psi\,\}$.

As one illustrative case from the proof, we show that:

$$\models_{\mathsf{par}} \{\,\phi\,\}\,\texttt{while}\ B\ \texttt{do}\ C\,\{\,\psi\,\} \quad \text{implies} \quad \vdash_{\mathsf{par}} \{\,\phi\,\}\,\texttt{while}\ B\ \texttt{do}\ C\,\{\,\psi\,\},$$

As the induction hypothesis for this case, we have that, for all assertions ϕ', ψ',

$$\models_{\mathsf{par}} \{\,\phi'\,\}\, C\, \{\,\psi'\,\} \quad \text{implies} \quad \vdash_{\mathsf{par}} \{\,\phi'\,\}\, C\, \{\,\psi'\,\}.$$

Suppose then that $\models_{\mathsf{par}} \{\,\phi\,\}\, W\, \{\,\psi\,\}$, where W abbreviates $\texttt{while}\ B\ \texttt{do}\ C$.

...continued on next slide

We shall show that

1. $\models_{\text{par}} \{\, \text{wp}(W,\psi),\ B \,\}\ C\ \{\, \text{wp}(W,\psi) \,\}$.
2. $\mathbb{Z} \models \phi \to \text{wp}(W,\psi)$.
3. $\mathbb{Z} \models \text{wp}(W,\psi) \land \neg B \to \psi$.

It then follows that we have the proof tree below:

$$
\dfrac{\dfrac{\begin{array}{c}\text{(from 1, by induction hypothesis)}\\ \vdots \\ \{\, \text{wp}(W,\psi),\ B \,\}\ C\ \{\, \text{wp}(W,\psi) \,\}\end{array}}{\{\, \text{wp}(W,\psi) \,\}\ W\ \{\, \text{wp}(W,\psi),\ \neg B \,\}}\ \text{(partial while)}}{\{\, \phi \,\}\ W\ \{\, \psi \,\}}\ \text{(by 2 and 3)}
$$

It remains to establish 1–3.

. . . continued on next slide

For 1, by the weakest precondition lemma (1), we have

$$\models_{\mathsf{par}} \{\, \mathsf{wp}(W, \psi) \,\}\; W \;\{\, \psi \,\} \quad .$$

Whence

$$\models_{\mathsf{par}} \{\, \mathsf{wp}(W, \psi),\; B \,\}\; W \;\{\, \psi \,\} \quad ,$$

which, because W is the same as $C; W$ when B is true, is equivalent to

$$\models_{\mathsf{par}} \{\, \mathsf{wp}(W, \psi),\; B \,\}\; C\,;\; W \;\{\, \psi \,\} \quad .$$

Whence, by the sequencing lemma (2), indeed:

$$\models_{\mathsf{par}} \{\, \mathsf{wp}(W, \psi),\; B \,\}\; C \;\{\, \mathsf{wp}(W, \psi) \,\} \quad .$$

. . . continued on next slide

For 2, we have assumed that $\models_{\text{par}} \{\, \phi \,\}\; W \;\{\, \psi \,\}$. So, by the weakest precondition lemma (2), $\mathbb{Z} \models \phi \to \mathrm{wp}(W, \psi)$, as required.

For 3, by the weakest precondition lemma (1), we have

$$\models_{\text{par}} \{\, \mathrm{wp}(W, \psi) \,\}\; W \;\{\, \psi \,\} \quad .$$

Whence

$$\models_{\text{par}} \{\, \mathrm{wp}(W, \psi),\ \neg B \,\}\; W \;\{\, \psi \,\} \quad ,$$

which, because W is the same as `skip` when B is false, is equivalent to

$$\models_{\text{par}} \{\, \mathrm{wp}(W, \psi),\ \neg B \,\}\; \texttt{skip} \;\{\, \psi \,\} \quad .$$

In other words, indeed, $\mathbb{Z} \models \mathrm{wp}(W, \psi) \wedge \neg B \to \psi$.

$\square$